We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

10

Evaluation of the Delta-Sigma modulator coefficients by MATLAB parallel processing

Michal Pavlik, Martin Magat, Lukas Fujcik and Jiri Haze
Brno University of Technology
Czech Republic

1. Introduction

The task of the ΔΣ modulator design is the fact that ΔΣ modulator is nonlinear discrete system. Thus, the calculation of the optimal transfer coefficients is difficult. There exist three main design approaches: utilization of the table values, calculation from signal transfer and noise transfer functions (*STF, NTF*) and by iteration methods. At first, it is necessary to define tests and test conditions for optimization of the modulator transfer coefficients. Test results are used for consequent optimization steps. Spectral analysis is used to calculation of the signal to noise ratio (*SNR*) of the ΔΣ modulator output. The Fast Fourier Transform (FFT) is used for calculations. Accuracy of the *SNR* calculation directly depends on number of the spectral lines of the input signal bandwidth. Unfortunately, increasing number of the spectral lines also leads to exponential increasing of time demand. The low frequency or band pass filter is used inside modulator structure. Due to *SNR* of modulator would be different for various frequencies of the input signal in input signal bandwidth. Logically the modulator *SNR* would be dependent on the input signal amplitude. It is crucial to get relevant test results to ensure appropriate test conditions and resolution.

It is possible to calculate coefficients of the ΔΣ modulator based on signal and noise transfer functions instead of utilizing of the table values. It allows calculating values of the ΔΣ modulator transfer coefficients. Nevertheless, the coefficients ensure modulator stable, they are not apparently optimal. We usually use interpolation methods to determinate optimal values of the transfer coefficients. However, the number of the interpolation steps issue appears at this point. If we suppose the second order CIDIDF ΔΣ modulator, we can optimize total eight coefficients. Next, if we use only 64 iteration steps to each of eight coefficients it leads to the total of 64^8 (approximately 3.10^{14}) combinations. It is also number of the necessary FFT analysis to calculate. In addition, if we would calculate with various frequencies and amplitudes of the input signal, the number of combinations would be higher. We can see that it is not possible to calculate each combination by using computing power of the common personal computers. That is why we are looking for faster calculation like another optimization methods.
There exist a lot of optimizing methods. We would like to deal with the aspects of mentioned application for optimal coefficients values calculation of the modulator ΔΣ,

namely for computers with one or more processor cores. Next, the possibility of the computation cluster using will be described and another parallelization methods and processes as well. General comparisons of each described parallelization methods will be introduced in this chapter.

2. The number of spectral components issue

2.1 SNR and THD calculation issue

There are two most important parameters defining the AD converter quality and area of the utilization: conversion rate and effective number of bits (*ENOB*). The dynamic parameters (including *ENOB*) are usually obtained for harmonic sinusoidal signal (IEEE, 2000), (Kester & Sheingold, 2004). We can write (Norsworthy et al. ,1997), (Geerts et al., 2002)

$$ENOB = \frac{SNDR_p - 1.76}{6.02} \tag{1}$$

where *SNDR* is signal to noise distortion ratio for sinusoidal signal with maximal amplitude. The *ENOB* parameter concerns the distortion due to nonlinear transfer characteristics and overload of the quantization stage. The *SNDR* is very important for ΔΣ modulators. Sometimes it is called *SINAD*. Therefore it must be calculated to obtain *ENOB* (Kester, 1999)

$$SINAD = 20\log\left(\frac{S}{N+D}\right) = -10\log\left[10^{-\frac{SNR}{10}} + 10^{-\frac{THD}{10}}\right] \tag{2}$$

where *S* is energy of the input signal, *N* is energy of the quantization noise, *D* is energy of the harmonic distortion, *SNR* is signal to noise ratio and *THD* is total harmonic distortion. The IEEE Std. 1241-2000 standard defines examination of the first 10 harmonic components. However the integrated circuits producers usually do not follow this definition, i.e. the Analog Devices company analyzes only first 6 harmonic components. The reason is very simple. When calculating *THD*, only first 5 harmonic components mainly influence this calculation. The error between calculations from first 10 or 5 harmonic components is only tenth of dB (Kester, 1999). The *THD* parameter is (Kester & Sheingold, 2004)

$$THD = 20\log_{10}\left(\frac{P_{sig}}{P_{noise} + P_{dis}}\right) = -10\log_{10}\sqrt{\sum_{i=2}^{n}\left[10^{-\frac{V_i}{20}}\right]^2} \tag{3}$$

where P_{dis} is energy of the input signal distortion and *V*i is amplitude of the *i*-th harmonic component. The analysis of the *THD* and *ENOB* is simple. Fig. 1 shows frequency spectrum

of the converter with sampling frequency of 100 MHz and input signal with frequency of 35 MHz. The first 10 harmonic components of signal fa are shown. Aliased harmonics of f_a fall at frequencies equal to

$$f_{hn} = \left| \pm Kf_s \pm nf_{in} \right| \tag{4}$$

where n is the order of the harmonic, and K = 0, 1, 2, 3,....

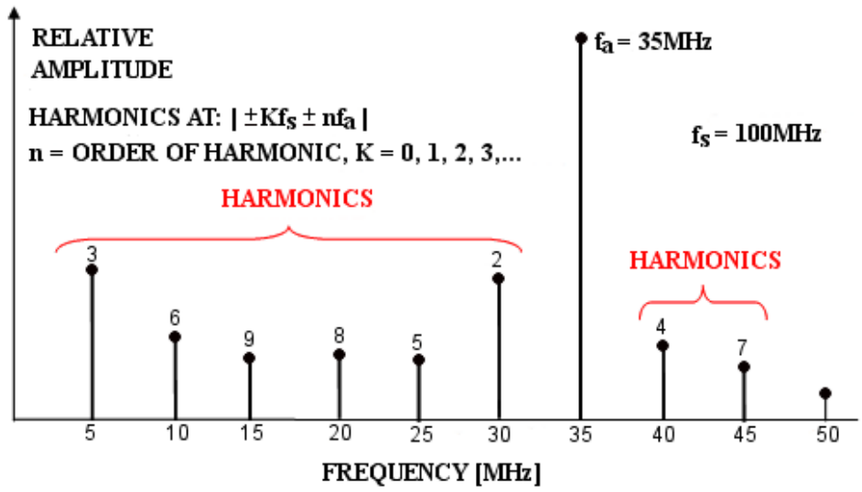

Fig. 1. Spectral analysis of the converter

It can be seen that for DFT (Discrete Fourier Transform) result 10.*i*, where *i* = 1,2,3,… of spectral components, the identification of the first 10 harmonic components is simple. The complicated situation is for mismatched spectral components and frequency of the harmonic components. The resulting error should be in tens of %. Nevertheless, when calculating *THD* it is possible to determine the number of spectral components in relation with input signal frequency to avoid the problem. The number of spectral components necessary for FFT (Fast Fourier Transform) is

$$M = D\left(\frac{f_s}{D(f_s, f_{in})}, 2^n \right) \tag{5}$$

where D is most common divisor. Unfortunately, the important disadvantage of the FFT algorithm is occasion of 2^n of spectral components.

The second parameter affecting *ENOB* of the AD converter is *SNR* (Kester, 1999)

$$SNR = \frac{P_{sig}}{P_{noise}} - 10 \log_{10}\left(\frac{f_s}{2.BW} \right) \tag{6}$$

where P_{sig} is energy of signal, P_{noise} is noise energy, f_s is sampling frequency and BW is bandwidth. Unfortunately this equation cannot be used for any case. The calculation error occurs for AD converters which spectral modulate quantization noise.

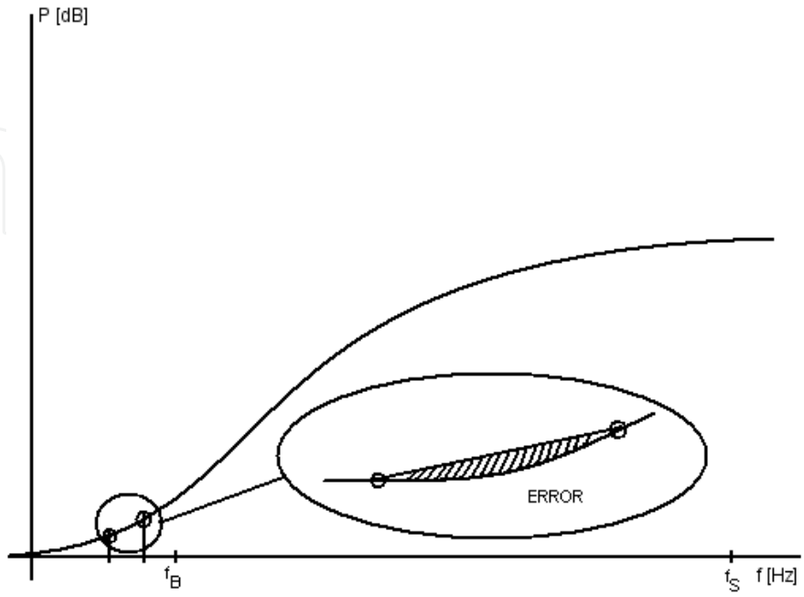

Fig. 2. The error of defining *SNR*

Several facts influence this error. There is mainly number of spectral components used for calculation, order modulator noise and oversampling ratio (*OSR*). It can be confirmed direct relation between growing number of spectral components and resulting accuracy of calculation.

The behaviour and function confirmation of ΔΣ modulators could be processed utilizing tools and scripts called SDtoolbox 2 (Brigati et al., 2004). It is very universal tool and the result of the calculation is value of *SNDR* (Malcovati et al., 2003). On the other hand, it is not able to differ contribution of particular errors on *spurious free dynamic range SFDR*

2.2 DFT leakage

The frequency analysis of the AD converter output signal should be done for calculation of both parameters (*SNR* and *THD*). It leads to calculation of DFT realized using FFT algorithm. However another problem occurs at this point. It is DFT leakage (Lyons, 2004). It is defined as energy distortion of one spectral component into its neighbour components. This situation arises when the ratio between frequency of sampling signal and input signal is not integer – Fig. 3. Nevertheless it is possible to set the frequency of input signal correctly during simulation. The AD converter must be able to process signals with any frequency in real situation.

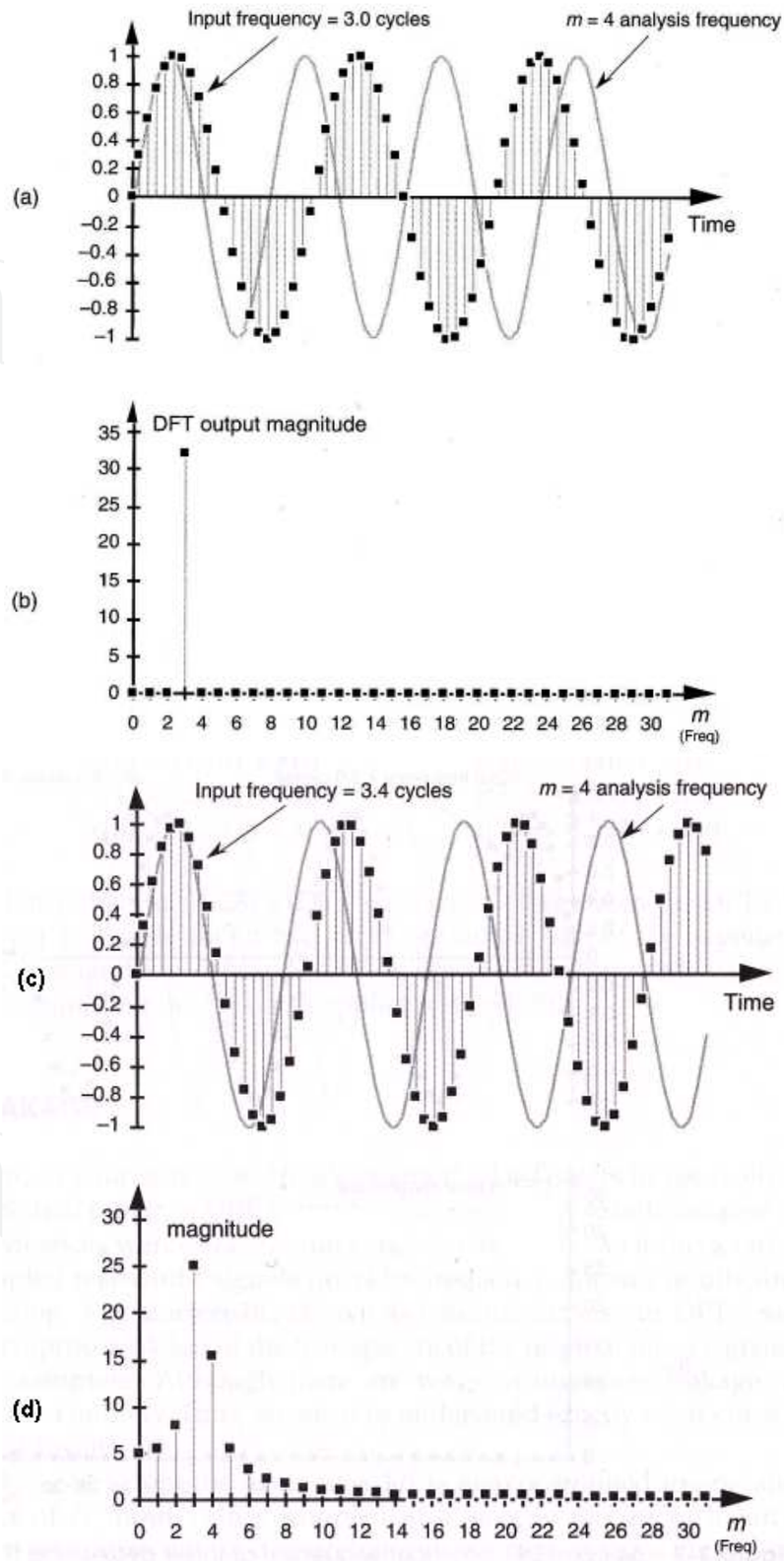

Fig. 3. Dependency of the DFT leakage

2.3 Computing time

The growing number of spectral components leads to the higher accuracy of *SNDR* calculation, but also grows computing time. This relation is exponential, but the deviation change caused by calculation decreases very fast.

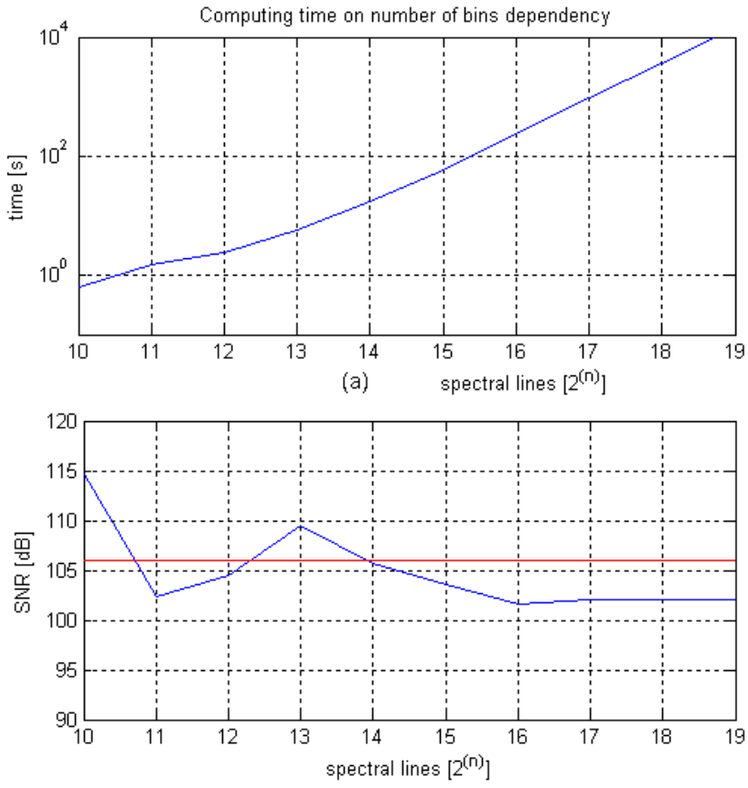

Fig. 4. Modulator *SNR* computing time consumption

The sufficient accurate result of simulation is obtained, when number of spectral components is higher than half of *OSR*.

3. Computing of the modulator transfer parameters

There exist three possibilities of determination of the ΔΣ modulator transfer parameters. They are:

- Utilization of table values,
- The calculation based on *STF* and *NTF*,
- Iteration methods.

The first method is useless due to its simplicity. The second is more complicated. It should be spited in two groups. One way uses fundamental behaviour of ΔΣ modulator with basic transform functions

$$STF = 1 \tag{7}$$

$$NTF = (z-1)^l \tag{9}$$

where l is order of the modulator.

The second way is utilization of table values of optimal transfer functions and transfer parameters calculation. This solution is universal and it should be applied on various types of DA modulators.

The third method is focused on observation of ideal ΔΣ modulator parameters by means of iteration. However, since the modulator is nonlinear system, the iteration is possible only by partial intervals. All appropriate constants must be iterated during transfer coefficients calculation. Fig. 5 shows the second order CIDIDF ΔΣ modulator, which were used in experiments.

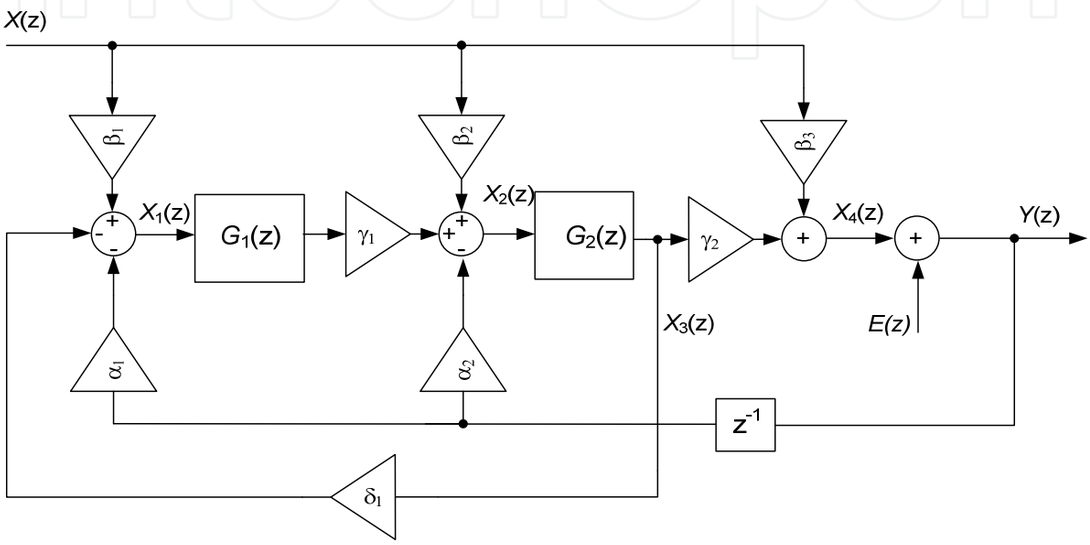

Fig. 5. Block scheme of the second order CIDIDF ΔΣ modulator

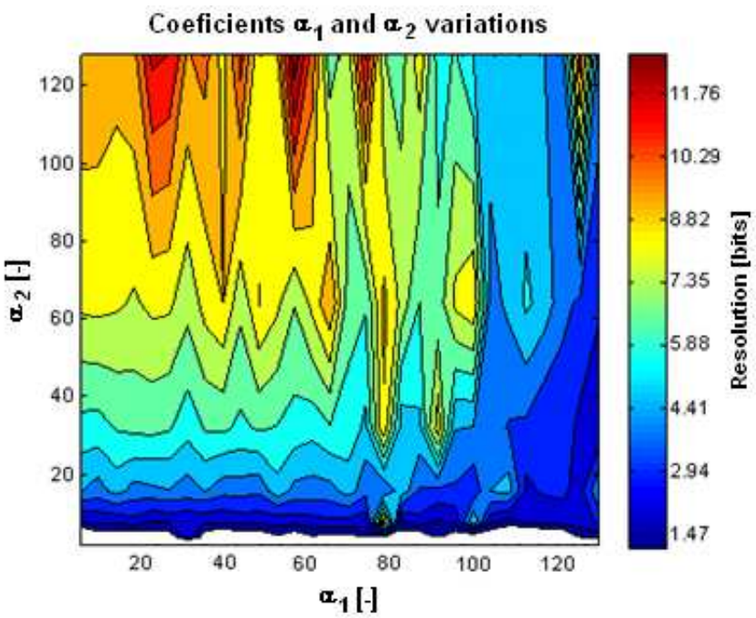

Fig. 6. *SNR* on coefficients α_1 and α_2 dependent

The input parameters are:

- *OSR*,
- bandwidth,
- limits of parameters,
- amplitude of the input signal.

It is possible to change eight parameters in this case. Their values affect each other. The example of simulation result for two parameters is shown in Fig. 6.

Consequently, it means that for iteration of i.e. 64x parameters, it is necessary to calculate $2{,}814.10^{14}$ times *SNR* of modulator. Therefore it is not possible to utilize this solution. The total computing time will take hundreds of years. That is why the optimization methods for iteration process must be used. One solution leads to several computing units utilization, which speed-up the calculation *n*-times. The second approach is genetic algorithm (GA) (Mitchell, 1996).

4. Computing cluster and its using in optimization methods

The aim of this chapter is not comparison of all computing parallelization methods. It describes the most useful method for our purpose. Since the computing tools for ΔΣ modulator simulation are created for MATLAB SIMULINK, we utilized this software.

There are many reasons why optimize methods, which use multi-results algorithm (e.g. GA, Particle Swarm Optimization (Kennedy & Eberhart, 1995), etc.). The first advantage is high efficiency of the solving of selected tasks accompanied with the fact that computing is very simple parallelizable as well. It gives a possibility to compute with multi-core systems if the algorithm is properly designed and parallelization is adequately processed. Additionally, the computing can be processed by any computer cluster that can be composed of many computers. It is simple and relatively cheap way, to enhance computing power and decrease the computing time.

4.1 Parallelization

There are many ways to parallelization of computing tasks in MATLAB. Unfortunately, methods like "parloop" or "matlab pool" are useable only for certain computing algorithms. Moreover, it speeds-up the computing minimally.

Another possibility of parallel computing is based on using of „Parallel computing toolbox"(PCT) (MATLAB, 2006). It enables parallel calculations on local station. Next method is utilization of „Distributed computing engine"(MDCE). It divides computing task into more computing stations. The main advantage of the MDCE against PCT is the fact that all parallel instances of the MATLAB are running and waiting for computing task instead of the PCT case, where MATLAB instances are started and stopped on request. If computing time is shorter than time needed to start MATLAB, the PCT method is useless. Moreover, using the PCT method in case of many quick tasks could bring significant delay during computing.

4.2 Computer cluster

The computer cluster was created to verify parallelization possibility of tasks which would be useful for the simulations of the ΔΣ modulator. The computer cluster was created and placed behind the Network Address Translation (NAT). The restriction was applied due to security reason. It is not necessary to connect computer cluster from outer network. If the situation is opposite the computer with main "job manager" would have public IP address to ensure that the "workers" will be able to connect to it from outer network. The block scheme of the computer connection is shown in Fig. 7.

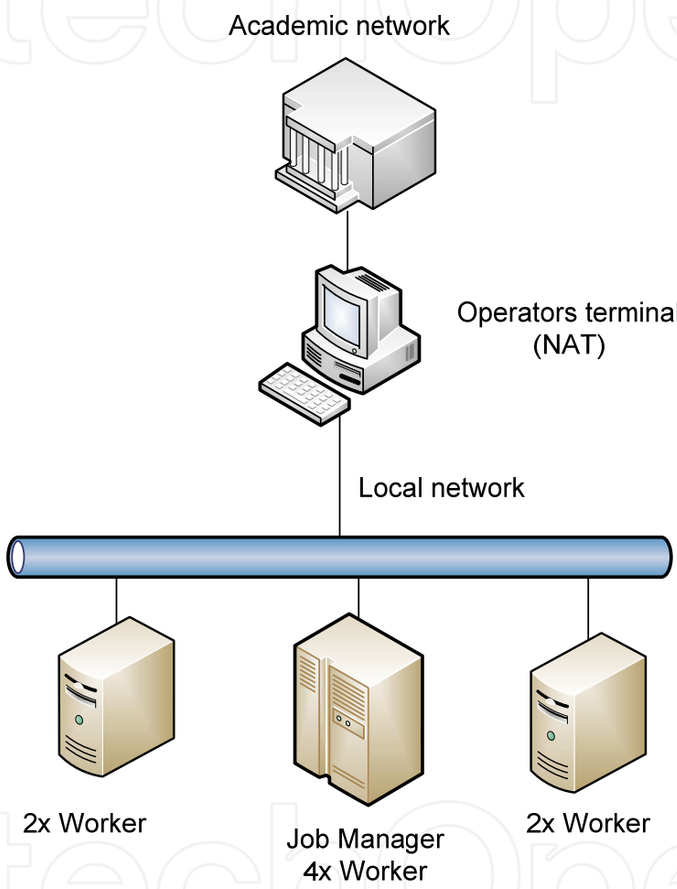

Fig. 7. The block scheme of the computer connection in the computer cluster

The crucial condition during MATLAB installation on computer connected into the network is proper MATLAB configuration on each connected computer. The MDCE could be executed from the system command line. First installation of the MDCE instance as "services" is necessary. The command "mdce install" serves for this purpose. Next the MDCE could be started by command "mdce start". Both commands should run from "bin" directory of the MDCE. It is usually "MATLAB\R2009b\toolbox\distcomp\bin". There is also "admin center" in same directory, which is executable in Windows operational system by command "admincenter.bat".

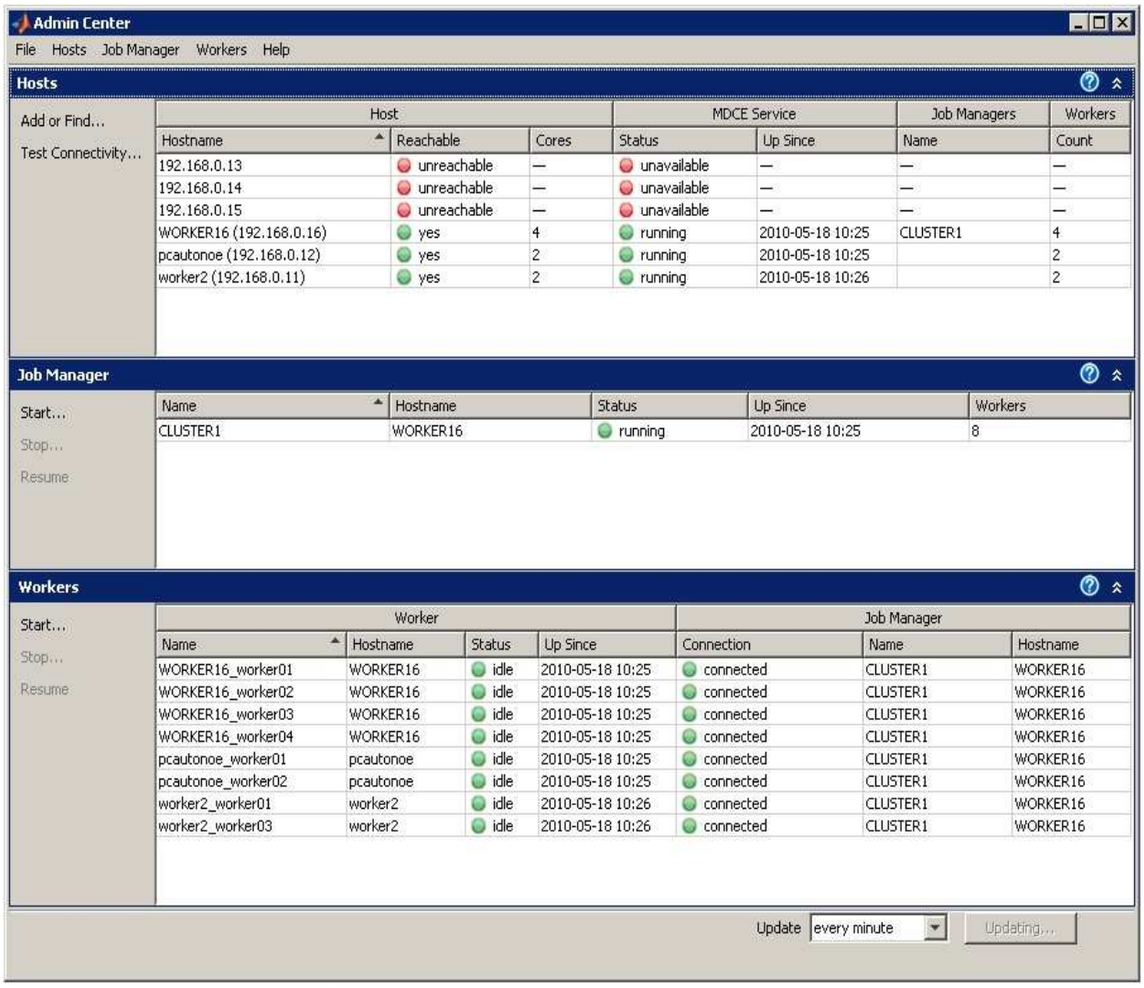

Fig. 8. The admin centre of the MDCE

Dialog window of the "admin centre" is divided into three parts placed underneath where the computer cluster is configured – Fig. 8. The connection of the each "worker" is controlled in the first part. There is displayed whether the "workers" are connected into the computer cluster and/or the MDCE runs there. The "job manager" is configured in the next part of the "admin centre". The "job manager" spreads computing tasks among the connected "workers".

Finally, the workers of the connected stations are executed in the third part of the "admin centre". It is an advantageous to run the same number of "workers" as a number of processor cores in the computer station.

Fig. 9 depicts the configuration of six computers in the cluster for our case. Three of them are temporary shut down. The figure shows the job manager "CLUSTER1" is configured on computer named "WORKER16". The running instances of the MDCE "workers" are doubled on computers "pcautonoe" and "wprker2" and four on computer "WORKER16". There are total 8 "workers" executed on three computers.

The block scheme of the MATLAB instances connected into the "job manager" is shown in Fig. 9.

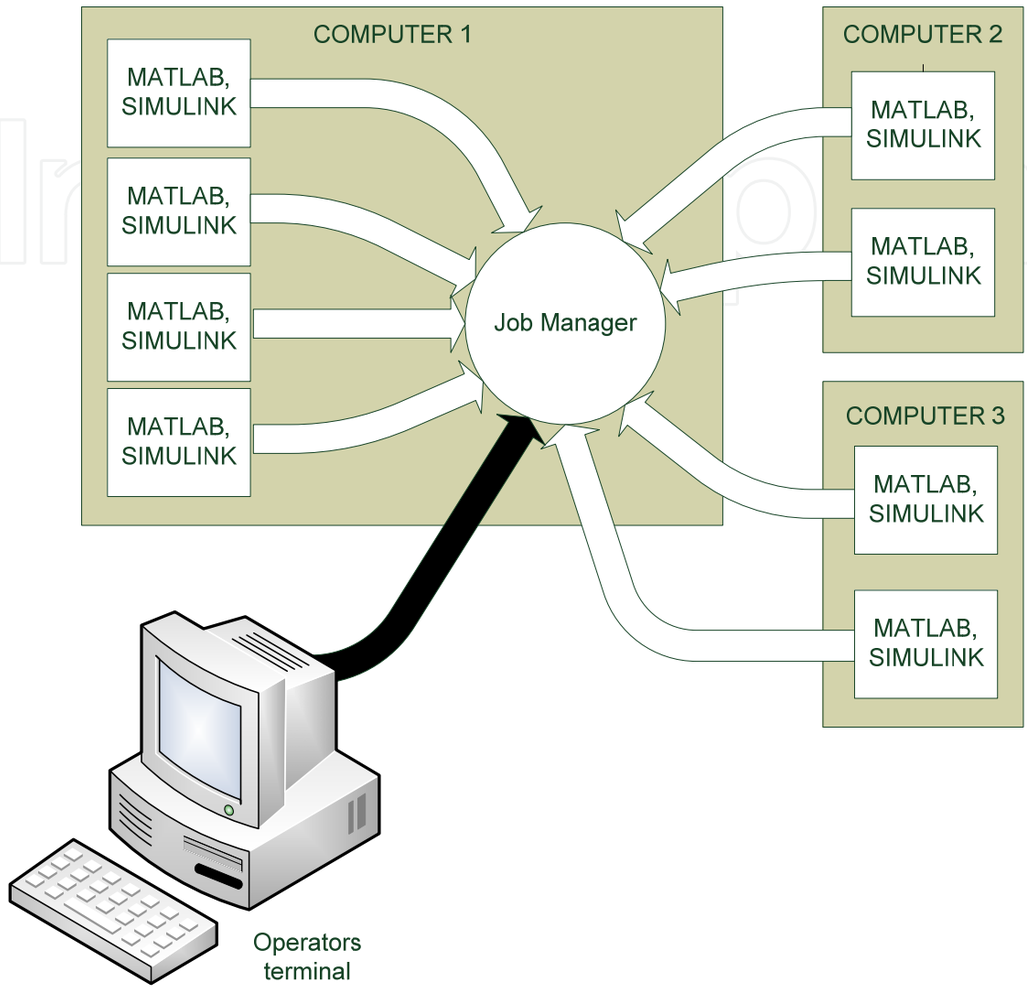

Fig. 9. The MATLAB instances connected into the "job manager"

There is also marked the connection of the operator computer in Fig. 9. The operator computer is sending calculation tasks. The MATLAB is configured to be as local "job manager". It is necessary to configure MATLAB to use cluster "job manager" to take advantage of the computer cluster - computer capacity. It is set in the bookmark "Parallel" of the MATLAB main menu. The new configuration of the "job manager" and IP address of the computer with running "job manager" of created computer cluster could be set in the "parallel" menu. The Fig. 10 shows the mentioned dialog box.

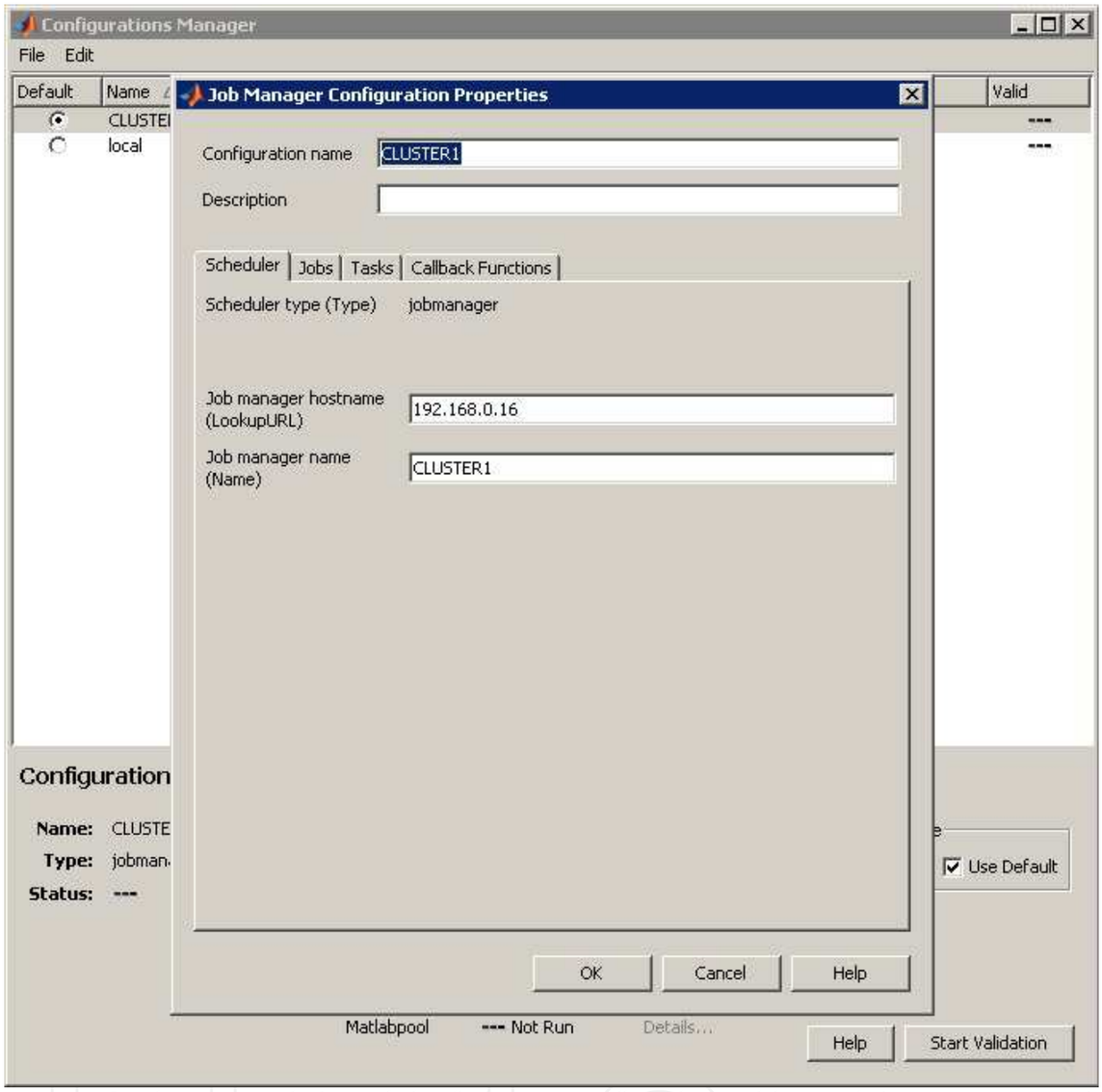

Fig. 10. Configuration dialog of the "job manager" and IP address of the computer with running "job manager" of created computer cluster

Next, the MATLAB must be configured for use of the new configuration to distribute computing task into the computer cluster.

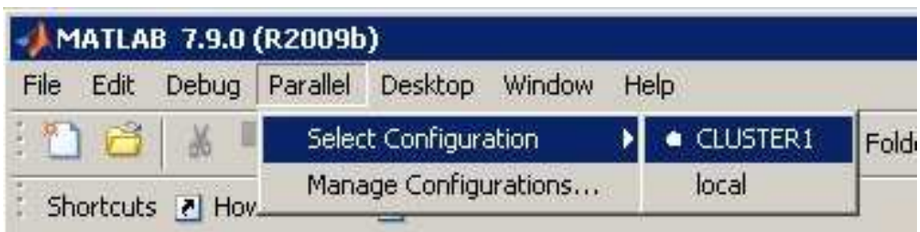

Fig. 11. MATLAB menu with the parallel computing items

4.3 Test of the computer cluster

The followed code was created to verify configuration and power of the computer cluster (soubor test_cluster_simulink.m , F_Test_Simulink.m). There is a function that uses the simulink for computing in the file F_Test_Simulink.m. The script described in test_cluster_simulink.m hundred times calculates function F_Test_Simulink in two configurations of the „job manager". In the first case the option is set as "local" (default settings) and in second case is set as "CLUSTER1" (the task is spread into the computer cluster). Computing time is measured in both cases.

```
test_cluster_simulink.m
clear all;
disp('start');

for i = 1:100
  p{i}=i;
end

startTime = tic;
a=dfeval(@F_Test_Simulink,p,'Configuration', 'CLUSTER1');
stopTime = toc(startTime);
fprintf('Cluster congiguration time: %g seconds.\n', stopTime);

startTime = tic;
a=dfeval(@F_Test_Simulink,p,'Configuration', 'local');
stopTime = toc(startTime);
fprintf('Local congiguration time: %g seconds.\n', stopTime);

disp('stop');
```

The result of this test is:

```
start
Cluster congiguration time: 14.9345 seconds.
Local congiguration time: 206.757 seconds.
stop
```

The test script was executed on the main computer of the computer cluster to obtain the most relevant result. It can be seen that the computing was 13-times faster in comparison with default settings. Note, it is remarkable result, especially considering the fact that the computation was calculated by eight computing threads. It is probably thanks to calculations processed without graphical interface (GUI) which requires the SIMULINK to be executed.

The function "dfeval" in mentioned code is used to parallelize the computation tasks. It is the simplest way how effectively executes tasks that have to be processed by PCT or MDCE. There are other methods to do it, but they are not useful for the ΔΣ modulator simulations. The main reason is that during parallelizing of GA task it is supposed all parameters of the functions are known before spreading computations of the criteria functions. The problem has to be solved in different way in difficult cases, especially in case of dynamic function.

5. Genetic algorithm

The GA is stochastic searching method based on the evolution algorithm. As a stochastic process the GA is always nondeterministic and cannot guarantee successful solution. The knowledge of the course of criteria (evaluative) function is not needed. It is main benefit of the GA technique. Next advantage of the GA is parallel computing possibility, since the algorithm operates with higher amount of results together. Finally, those are the main reasons why the GA was chosen and used for ΔΣ modulator coefficients evaluation.

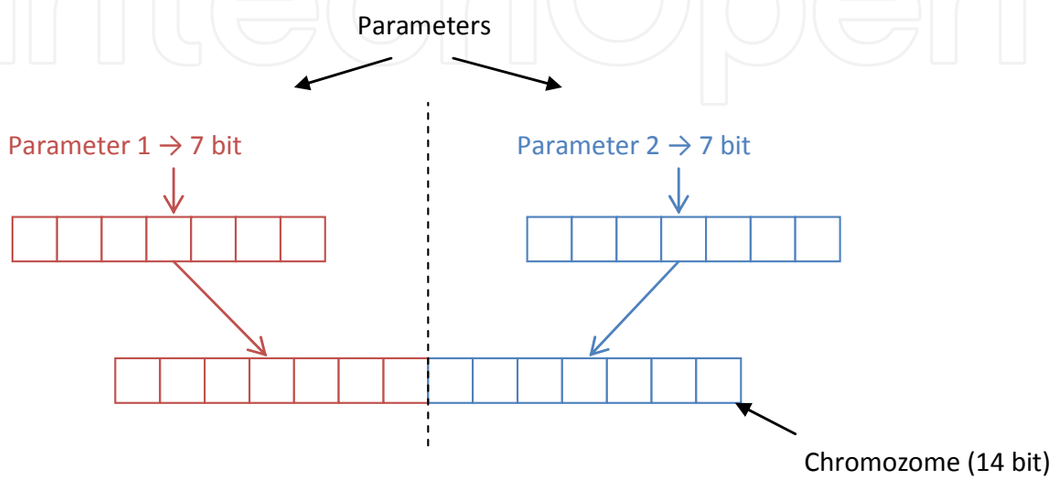

Fig. 12. Parameter coding

5.1 Parameters coding

The GA works with more results (subjects) which are collected into one generation. The subject represents sequence of bits, which is called chromosome. Parameters of the result are optimized and coded as a sequence of bits and put into the chromosome like a gene. Coding of the result parameters is shown in Fig. 12. Parameter coding is very interesting and provides coding also for unordinary types of parameters which would be difficult expressed by number g.e. smell or light colour.

5.2 Description of the genetic algorithm

The GA can be dividend into the six steps:

- Initialization of the starting population
- Coding of the solution parameters
- Gene creating from chromosome
- Subject evaluating of the population by criteria function
- Selecting of the best evaluated subjects
- Creating of the next generation based on the recombination and mutation of the selected subjects

Typical GA processing could be dividend into the three basic stages: Initialization, Reproduction and Exchange of the generations.

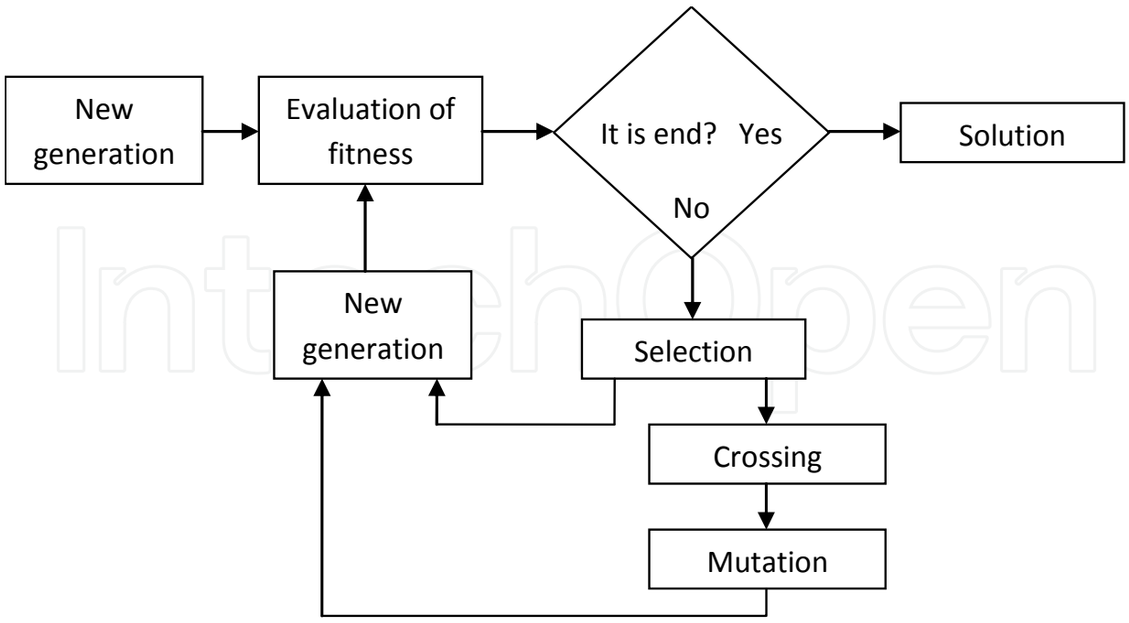

Fig. 13. Flowchart of the GA

The fundamental GA flowchart is shown in the Fig. 13. The first generation is filled by defined quantity of the randomly generated and coded unique subjects during the initialization. Each of the generated subjects represents one solution. The generated subjects are used as a new generation and consequently, each subject is evaluated by criteria function.

The new generation from older one is created during the reproduction phase. The reproduction means that the individual pair is selected. The selected pair serves like parents. Parents are hybridized and muted. They produce new pair called descendants. Consequently the descendants are placed into the new generation. Selection, hybridizing and mutations have to be processed until the sufficient amount of the descendants is generated for filling of the new generation.

5.3 Selection

The selection starts by the criteria function evaluating of the subjects. It uses results of the criteria function for each subject to determine the subject effectiveness. Nevertheless, selection is not only choosing the best subject, because the best subject need not be close to the optimal solution. The different selecting strategies are used depending on the concrete task. The most frequently used strategies are strategy of concurrent fight or tournament.

5.4 Hybridizing

The two parents are used to obtain two new descendants creating in operation of hybridizing. Many hybridizing methods are developed. One of the simplest is one-point hybridizing. The one-point hybridizing method is depicted in Fig. 14. It selects randomly place where the chromosomes of the parents are swapped.

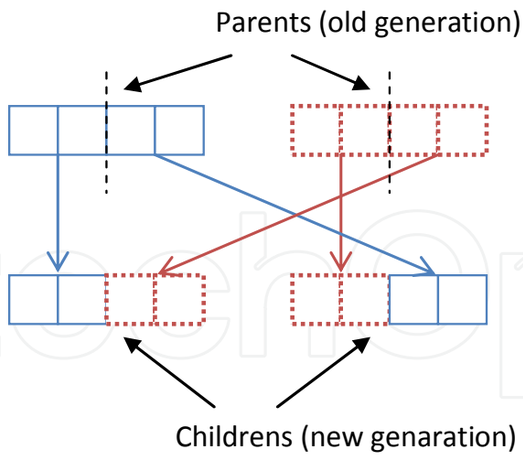

Fig. 14. Block scheme of the hybridizing process

5.5 Mutation

The chromosome is randomly chosen and arranged. The random bit is selected and inverted in the randomly selected chromosome. Example of the mutation is shown in Fig. 15.

5.6 Finalization of the genetic algorithm

The last step of the GA calculation is its finalization. The most frequently used method is displaying of the best searched solution after the defined number of GA runs.

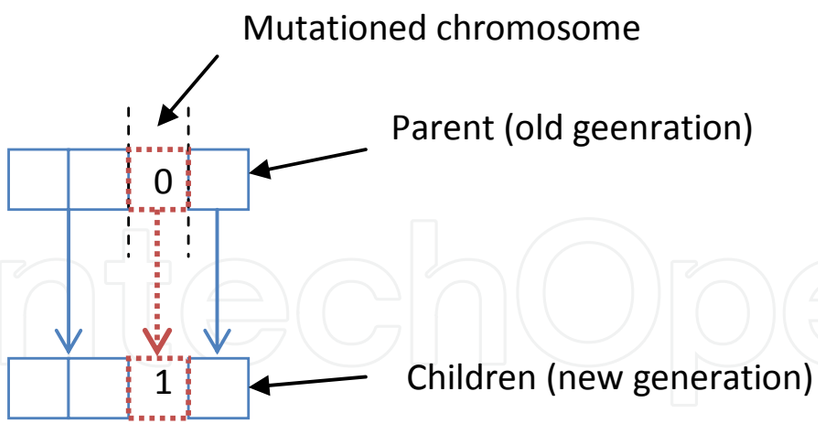

Fig. 15. The process of mutation

If the number of the algorithm solutions is not sufficient the possibility that the optimal solution would not be found exists. Alternative frequently finalizing method to terminate the GA algorithm is based on the computing termination when the solution with defined error is found. Since the GA is stochastic, the various results could be found. It is a serious problem of the GA. Due to the adequate number of the calculation runs and parameters for hybridizing and mutation have to be set as well.

6. Conclusion

The most important parameters, which affect this process, are conversion rate and effective number of bits (*ENOB*). The *ENOB* influences another features of the ΔΣ modulator such as signal to noise distortion ratio (*SNDR*) and total harmonic distortion (*THD*).
There are three methods of coefficients calculation – utilization of table values, the calculation based on signal and noise transfer function (*STF*, *NTF*) and iteration methods.
The article presents problems arising during MATLAB simulation of the ΔΣ modulator behaviour. It has been discussed the problem of finding of optimal spectral components number. Next, there have been depicted methods of determination of ΔΣ modulator transfer coefficients. The genetic algorithm has been presented in more details as one of the solution possibilities. The calculations require a lot of time**.** That is why the computer cluster has been made and its configuration and utilization have been presented. It has been shown how to find the optimal solution for certain task.

7. References

Brigati et al. (2004). A FourthOrder Single Bit Switched Capacitor ΣΔ Modulator for Distributed Sensor Applications; IEEE Transactions on Instrumentation and Measurement, Vol. 53, Issue 2, 2004, pp. 266 G270

Geerts et al. (2002) Design of Multi-Bit Delta-Sigma A/D Converters, The Springer International Series in Engineering and Computer Science, Vol. 686, 2002, 240 p., Hardcover ISBN: 978-1-4020-7078-5

IEEE (2000). IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters, IEEE 1241-2000

Johns & Martin (1997). Analog integrated circuit design; publisher John Wiley & Sons, Inc., USA; ISBN:0-471-14448-7

Kennedy & Eberhart (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942-1948.

Kester & Sheingold (2004). Chapter 5: Testing Converters, Analog Devices

Kester (1999). Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor, Analog Devices

Lyons (2004). Understanding Digital Signal Processing (2nd Edition), Prentice Hall PTR, Upper Saddle River, NJ, 2004

Malcovati et al. (2003). Bahavioral modelling of switched-capacitor Sigma–Delta modulators; IEEE Trans. Circuits Syst. I, vol. 50, no. 3, pp. 352–364, Mar. 2003.

MATLAB (2006) Parallel Computing Toolbox 4.3, MathWorks

Mitchell (1996). An Introduction to Genetic Algorithms. Cambidge, MA: MIT Press 1996

Norsworthy et al. (1997). Delta-Sigma Data Converters, Piscataway NJ, IEEE Press, 1997, 476 pages, ISBN 0-7803-1045-4

Roberts (2008). Test Methods For Sigma-Delta Data Converters and Related Devices; Proceedings of the 21st annual symposium on Integrated circuits and system design; publisher ACM New York, USA; ISBN:978-1-60558-231-3

Strle (2008). Efficient Testing of Σ-Δ A/D Converters; proceedings of 15th IEEE International Conference on Electronics, Circuits and Systems, 2008, ISBN:978-1-4244-2181-7, pp 1225-1228

Van de Plassche (2003). CMOS Integrated Analog-to-Digital and Digital -to-Analog Converters, 2nd Edition, publisher Kluwer Academic Publishers Dordrecht, Netherlands; ISBN:1-4020-7500-6

Zaplatílek & Doňar (2003). MATLAB pro začátečníky; publisher BEN Technická literatura, Praha, Czech republic; ISBN:80-7300-095-4

Zaplatílek & Doňar (2004). MATLAB tvorba uživatelských aplikací; publisher BEN Technická literatura, Praha, Czech republic; ISBN:80-7300-133-0

Zaplatílek & Doňar (2006). MATLAB začínáme se signály; publisher BEN Technická literatura, Praha, Czech republic; ISBN:80-7300-200-0

Matlab - Modelling, Programming and Simulations

Edited by Emilson Pereira Leite

ISBN 978-953-307-125-1
Hard cover, 426 pages
Publisher Sciyo
Published online 05, October, 2010
Published in print edition October, 2010

This book is a collection of 19 excellent works presenting different applications of several MATLAB tools that can be used for educational, scientific and engineering purposes. Chapters include tips and tricks for programming and developing Graphical User Interfaces (GUIs), power system analysis, control systems design, system modelling and simulations, parallel processing, optimization, signal and image processing, finite different solutions, geosciences and portfolio insurance. Thus, readers from a range of professional fields will benefit from its content.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Michal Pavlik, Lukas Fujcik, Martin Magat and Jiri Haze (2010). Evaluation of the Delta-Sigma Modulator Coeficients by MATLAB Parallel Processing, Matlab - Modelling, Programming and Simulations, Emilson Pereira Leite (Ed.), ISBN: 978-953-307-125-1, InTech, Available from: http://www.intechopen.com/books/matlab-modelling-programming-and-simulations/evaluation-of-the-delta-sigma-modulator-coeficients-by-matlab-parallel-processing

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen